10/523798

Office de la propriété intellectuelle du Canada

Un organisme d'Industrie Canada

Canadian Intellectual Property Office

An Agency of Industry Canada

PCT/CA 03/012154

10 SEPTEMBER 2003 10·09·03

Bureau canadien des brevets

Certification

Canadian Patent Office

REC'D 25 SEP 2003

Certification WIPO PCT

La présente atteste que les documents ci-joints, dont la liste figure ci-dessous, sont des copies authentiques des documents déposés au Bureau des brevets.

This is to certify that the documents attached hereto and identified below are true copies of the documents on file in the Patent Office.

Specification and Drawing, as originally filed, with Application for Patent Serial No: **2,397,424**, on August 9, 2002, by **MOHAMMED LAMINE KHERFI, DJEMEL ZIOU AND ALAN BERNARDI**, for "Content-Based Image Retrieval Using Positive and Negative Examples".

RIORITY DOCUMENT
BMITTED OR TRANSMITTED IN
COMPLIANCE WITH
RULE 17.1(a) OR (b)

Agent certificateur/Certifying Officer

September 10, 2003

Date

Canada

(CIPO 68)
04-09-02

OPIC   CIPO

ABSTRACT OF THE DISCLOSURE

Although negative example can be highly useful to better understand the user's' needs in content-based image retrieval, it was considered by few authors. We address some issues related to the combination of positive and negative examples to perform a more efficient image retrieval. A relevance feedback approach that uses positive example to perform generalization and negative example to perform specialization is described herein. In this approach, a query containing both positive and negative example is processed in two steps. The first step considers positive example only in order to reduces the set of images participating in retrieval to a more homogeneous subset. Then, the second step considers both positive and negative examples and, acts on the images retained in the first step. Mathematically, relevance feedback is formulated as an optimization of intra and inter variances of positive and negative examples.

1

TITLE OF THE INVENTION

Content-Based Image Retrieval Using Positive and Negative Examples

FIELD OF THE INVENTION

[0001] The present invention relates to image retrieval. More specifically, the present invention is concerned with content-based image retrieval using positive and negative examples.

BACKGROUND OF THE INVENTION

[0002] With advances in the computer technologies and the advent of the World-Wide Web, there has been an explosion in the quantity and complexity of digital data being generated, stored, transmitted, analyzed, and accessed [30]. These data take different forms such as text, sound, images and videos.

[0003] The huge number of images available renders it necessary to develop systems for efficient image retrieval which can help users locate the needed images in a reasonable time. These systems use many attributes of the images, such as the presence of a particular combination of colors or the depiction of a particular type of event [19]. Such attributes may either be derived from the content of the image or from its surrounding text and data. This leads to various approaches in image retrieval such as content-based techniques and text-based techniques. A good survey on the techniques used in image retrieval can be found in [2], [19] and [29].

2

[0004]   If we focus on content-based image retrieval, we can distinguish many ways of formulating queries. Early systems such as QBIC [18] ask the user to select image features such as color, shape, or texture. Other systems like BLOBWORLD [4] require the user to provide a weighted combination of features. However, it is generally difficult to directly specify the features needed for a particular query, for several reasons. First, because not all users understand the image vocabulary (e.g. contrast, texture, color) needed to formulate a given query. Second, even if the user is an image specialist, it is not easy to translate the images the user has in mind into a combination of features.

[0005]   An alternative approach is to allow the user to specify the features and their corresponding weights implicitly via a visual interface known as "query by example". The user can choose images that will participate in the query and weight them according to their resemblance to the images sought. The results of the query can then be refined repeatedly by specifying more relevant images. This process, called Relevance Feedback (RF), is defined in [28] as the process of automatically adjusting an existing query using information fed back by the user about the relevance of previously retrieved documents. Relevance feedback is used to model the user subjectivity in several stages. First, it can be applied to identify the ideal images that are in the user's mind. At each step of the retrieval, the user is asked to select a set of images which will participate in the query; and to assign a degree of relevance to each of them. This information can be used in many ways in order to define an analytical form representing the query intended by the user. The ideal query can be defined independently from previous queries, as in [25]. It can also depend on the previous queries, as in the query point movement method where the ideal query point is moved towards positive example and away from negative example [8]. Examples of this method can be found in [28] and [20].

3

The computation of the ideal query may be unnecessary in certain cases such as the current work, where we automatically integrate what the user is looking for into the similarity measures without the need to identify any ideal point.

[0006]   Relevance feedback allows also to capture the user's needs by assigning a degree of importance (e.g. weight) to each feature or by transforming the original feature space into a new one that best corresponds to the user's needs and specificities. This is done by enhancing the importance of those features that help in retrieving relevant images and reducing the importance of those which do not. Once the importance of each feature is determined, the results are applied to define similarity measures which correspond better to the similarity intended by the user in the current query.

[0007]   The operation of attributing weights to features can also be applied to perform feature selection, which is defined in [31] as the process of choosing a subset of features by eliminating redundant features or those providing little or no predictive information. In fact, after the importance of each feature is determined, feature selection can be performed by retaining only those features which are important enough; the rest will be eliminated. By eliminating some features, retrieval performance can be improved because in a low-dimension feature space, it is easier to define good similarity measures, to perform retrieval in a reasonable time, and to apply effective indexing techniques [19].

[0008]   While many studies have focused on how to learn from user interaction in relevance feedback [26, 30, 25, 23, 3, 9, 17], few of them evoked the relevance of negative example [21, 7, 20, 5]. Negative example can be useful for query refinement since it allows to determine the images the user doesn't want in order to discard them. In [7], the authors show that when they

4

use only positive feedback, the major improvement occurs at the first feedback step; and that the improvement with positive and negative feedback is remarkable for the four first steps where the results continuously get better. We also note that relevance feedback with negative example is useful to reduce noise (undesired images that have been retrieved) and to decrease the miss (desired images that have not been retrieved).

[0009]   Relevance feedback using positive example has been considered by many authors [1, 30, 26, 25, 24, 9, 17]. Ishikawa et al. [25] define a quadratic distance function for comparing images. Consider that we have a query consisting of N images, each image is represented by an I-dimension feature vector $\vec{x}_n = [x_{n1},...,x_{nI}]^T$, where T denotes matrix transposition. Consider also that the user associates each image participating in the query with a degree of relevance π_n which represents its degree of resemblance with the sought images. The authors compute two parameters, namely the ideal query $\vec{q} = [q_1,...,q_I]^T$ and the ellipsoid distance matrix W, that minimize the quantity D given in Equation (1), which represents the global distance between the query images and the ideal query.

$$D = \sum_{n=1}^{N} \pi_n (\vec{x}_n - \vec{q})^T W (\vec{x}_n - \vec{q})$$

(1)

[0010]   In [26], Rui. et al. decompose each image into a set of I features, each of which is a vector of reals. Let $\vec{x}_{ni}$ be the ith feature vector of the nth query image and π_n the degree of relevance assigned by the user to the nth image. Assume also that the query consists of N images. The authors compute, for each feature i, the ideal query vector $\vec{q}_i$, a matrix W_i and scalar weight u_i which minimize the global dispersion of the query images given by

5

Equation (2). By minimizing the dispersion of the query images, they try to enhance the concentrated features, i.e., features for which example images are close to each other.

$$J = \sum_{i=1}^{I} u_i \sum_{n=1}^{N} \pi_n (\vec{x}_{ni} - \vec{q}_i)^T W_i (\vec{x}_{ni} - \vec{q}_i)$$

(2)

[0011] In [27], the authors propose to use the same model but with negative degrees of relevance assigned to negative example images. This consideration leads to neglect the relevant features of negative example, so that negative example will be confused with positive example.

[0012] If we consider the way negative example was interpreted in previous studies, we can distinguish two categories of models. In the first category, the positive example images are selected by the user; however, the negative example images are chosen automatically by the retrieval system among those not selected by the user. In the second category, both positive and negative example images are chosen by the user.

[0013] We will begin by examining the first category, which includes the work of Müller et al. [7]. Concerning the initial query, they propose to enrich it by automatically supplying non-selected images as negative example. For refinement, they select the top 20 images resulting from the previous query as positive feedback. As negative feedback, they choose four of the non-returned images. Their system performs refinement through several feedback steps; in each step, it tries to move the ideal query towards the positive example and away from the negative example. This is done by using the Rocchio formula [12]:

6

$$Q = \frac{\alpha}{n_1} \sum_{i=1}^{n_1} R_i - \frac{\beta}{n_2} \sum_{i=1}^{n_2} S_i$$

(3)

where Q is the ideal query, n_1 and n_2 are the numbers of positive and negative images in the query respectively, and R_i and S_i are the features of the positive and negative images respectively. α and β determine the relative weighting of the positive and negative examples. They adopt the values $\alpha = 0.65$ and $\beta = 0.35$ used by some text-retrieval systems [7]. We notice that in this category of models, since the system selects negative example images automatically, it is very important to use the right images, otherwise this can destroy the query. Indeed, if the system chooses as negative example some images which should rather be considered as positive example, then the relevant features of these images will be discarded, and this will mislead the retrieval process.

[0014] The second category includes the work of Vasconcelos and Lippman [20] where they propose a Bayesian model for image retrieval, operating on the assumption that the database is constituted of many image classes. When performing retrieval, they support image classes that assign a high membership probability to positive example images, and penalize image classes that assign a high membership probability to negative example images. We notice that the authors consider that the positive and the negative examples have the same relative importance.

[0015] In [23, 24], Picard et al. organize the database images into many hierarchical trees according to individual features such as color and texture. When the user submits a query, they perform comparison using each of the trees, then they combine the resulting sets by choosing those image sets which most efficiently describe positive example, with the condition that these sets don't describe negative example well.

[0016] In [21], Belkin et al. use a Bayesian probabilistic model in which they assume that the relevant features of positive example are good, whether or not they are relevant to negative example. Their interpretation of negative example is that the context in which positive example appears is inappropriate to the searcher's problem. They propose to increase the (positive) weight of the relevant features of positive example (irrespective of their appearance in negative example); and to enhance (with negative weights) the relevant features of negative example which don't appear in positive example. Enhancing important features of positive example which also appear in negative example, can mislead the retrieval process, as will be discussed hereinbelow.

[0017] In [6, 5], the authors consider that the image database is made up of relevant images, among which the user chooses positive example, and non-relevant images, among which the user chooses negative example. They use a probabilistic model in which they try to estimate the distribution of relevant images and to simultaneously minimize the probability of retrieving non-relevant images.

OBJECTS OF THE INVENTION

[0018] An object of the present invention is therefore to provide improved content-based image retrieval using positive and negative examples.

[0019] Other objects, advantages and features of the present invention will become more apparent upon reading of the following non-restrictive description of preferred embodiments thereof, given by way of example only with reference to the accompanying drawings.

8

[0020]   It is to be noted that various documents are referred to herein.   These documents are hereby incorporated by reference in their entirety.

BRIEF DESCRIPTION OF THE DRAWINGS

[0021]   In the appended drawings:

[0022]   Figure 1 illustrates precision-scope curves for two cases: negative example in two steps and negative example in one step;

[0023]   Figure 2 illustrates a positive average; a negative average, and the overall query average;

[0024]   Figure 3 illustrates the minimization of the global dispersion leading to neglect the relevant features of negative example;

[0025]   Figure 4 illustrates the minimization of the dispersion of positive example, the minimization of negative example and the minimization of the distinction between them;

[0026]   Figure 5 illustrates an example of the images a user has to choose from;

[0027]   Figure 6 illustrates a series of choice the user may make regarding any of the seen pictures;

[0028]   Figure 7 illustrates the decomposition of the HIS color space into a set of subspaces and the computation of each subspace's histogram;

9

[0029]   Figure illustrates and example of the results of a search done with positive example only;

[0030]   Figure 9 illustrates an example of refinement with both positive and negative examples;

[0031]   Figure 10 illustrates precision-scope curves for retrieval with positive example and refinement with negative example; and

[0032]   Figure 11 is a table on the number of iterations needed to locate a given category of images in two cases: using positive example only and using both positive and negative examples.

DESCRIPTION OF THE EMBODIMENT

[0033]   When an image retrieval system returns the results of a given query, we often encounter two problems: noise and miss. Noise consists of retrieved images which don't correspond to what the user wants. Miss is the set of images corresponding to what the user wants which have not been retrieved. These two problems (noise and miss) occur because of imperfections at different levels. For the user, it may not be easy to formulate an adequate query using the available images, either because none of them correspond to what the user wants or because the user lacks knowledge of imagery details such as the meaning of features. For the retrieval system, it may be difficult to translate the user's needs and specificities in terms of image features and similarity measures. However, the use of negative example in query formulation and refinement can help to reduce noise and miss. Indeed, after the results of a given query are obtained, the user can maintain the positive example images and enrich the query by including some undesired images as negative

example. This implies that images similar to those of negative example will be discarded, thus reducing noise. At the same time, the discarded images will be replaced by others which would have to resemble better what the user wants. Hence, the miss will also be decreased. Furthermore, the user can find, among the recently retrieved images, more images that resemble what the user needs and use them to formulate a new query. Thus, the use of negative example would help to resolve what is called the page zero problem, i.e., that of finding a good query image to initiate retrieval. By mitigating the page zero problem, the retrieval time will be reduced and the accuracy of the results will be improved [19]. We also note that relevance feedback with negative example is useful when, in response to a user feed-back query, the system returns exactly the same images as in a previous iteration. Assuming that the user has already given the system all the possible positive feedback, the only way to escape from this situation is to choose some images as negative feedback [20].

[0034]   Now, let us examine how negative example can be interpreted and how this could be applied to relevance feedback. Existing systems consider negative example in two ways.

[0035]   Some systems like [23] consider it at image level. They search for the set of images similar to positive example, then they search for the set of images similar to negative example; and finally they manipulate the two sets in order to obtain the set of images that they return to the user.

[0036]   Other systems such as the one in [21] consider the negative example at feature level. They try to identify and enhance the features which help to retrieve images that are at the same time similar to positive example but not similar to negative example.

11

[0037]   In what follows, we will consider the negative example from the feature point of view, and use it to identify the most discriminating features according to the user-given query.

[0038]   In knowledge discovery in databases, most learning-from-example algorithms partition the set of examples into positive and negative subsets. They perform generalization using positive example and specialization using negative example [22]. These algorithms try to extract decision rules including characteristic rules and discrimination rules. A characteristic rule of a set is an assertion which characterizes a concept satisfied by all or most of the members of this set. For example, the symptoms of a specific disease can be summarized by a characteristic rule. A discrimination rule is an assertion which discriminates a concept of the target set from the rest of the database [10]. For example, to distinguish one disease from others, a discrimination rule should summarize the symptoms that discriminate this disease from others.

[0039]   Before we see how to apply the above-described principle to image retrieval, let us make the following hypothesis. We always assume that positive and negative examples possess some relevant features that are discriminant, i.e., relevant to either positive or negative example or to both but whose values are not the same in positive and in negative examples. In other words, we exclude the case in which the relevant features of positive example are the same as those of negative example, with similar values. In such a case, the query is ambiguous. The system rejects it and ask the user to specify a new one.

[0040]   Now, a first idea to apply the above-described principle to image retrieval would be the following. First, characteristic rules can be extracted from positive example images by the identification of their relevant

12

features. More importance must be given to such features in the retrieval process and images enhancing them should be retrieved. Second, discrimination rules can be extracted from the difference between positive example and negative example. Relevant features whose values are not common to positive and negative examples are good discriminators, and hence must be given more importance; conversely, common features are not good discriminators, and must be penalized. However, by applying this principle in this manner, we may mislead the retrieval process by neglecting certain relevant features of positive and negative examples, as explained below.

[0041] To make this idea clearer, let us first define what we mean by a relevant feature. We consider that a given feature is relevant if it helps to retrieve the images being sought. In our case, this will depend on two factors.

[0042] First, the relevance can be considered with respect to the query. A feature relevant to the query is a feature which is salient in the majority of the query images. As we hereinbelow, we will consider every feature whose values are concentrated in the query images, and which discriminates well between positive and negative examples, as relevant to the query.

[0043] Second, the relevance of a feature can be considered with respect to the database. If a given feature's values are almost the same for the majority of the database images, then this feature is not relevant since it doesn't allow to distinguish the sought images from the others; and vice versa. To illustrate, consider a database in which each image contains an object with a circular shape but the color of the object differs from one image to another. In such a database, the feature Shape is not interesting for retrieval since it doesn't allow to distinguish between desired and undesired images; however, the feature Color is interesting. In other words, a feature in term of which the

13

database is homogeneous is not relevant for retrieval; whereas, a feature in term of which the database is heterogeneous is relevant.

[0044] Now, let us analyze the consequences of neglecting features whose values are common to both positive and negative examples. In fact, this depends on the nature of the database. If the database is homogeneous in terms of such features, then neglecting them will not be disastrous since they are not relevant to the database. On the other hand, if the database is heterogeneous in terms of these features, then neglecting them will lead the system to retrieve many undesired images and to miss many desired images.

[0045] To develop a solution that works for any query, it is clear that common features should be considered. However, in some cases, there are not enough common features to be considered alone at a given moment; they must rather be considered together with other features.

[0046] According to an aspect of the present invention, a retrieval in two steps is proposed. A first step serves to reduce the heterogeneity of the set of images participating in the retrieval by restricting it to a more homogeneous subset according to positive example relevant features (and thus according to common features also). In this first step, we enhance all the relevant features of positive example. We rank the database images according to their resemblance to positive example and then retain only the Nb_1 top-ranked images, where Nb_1 is a predetermined number.

[0047] Only images retained in the first step will participate in the refinement performed in a second step, where we enhance the discrimination features, i.e., those whose values are not common to positive and negative examples. In this second step we rank the candidate images according to their

14

similarity to positive example and dissimilarity to negative example, and return to the user only the Nb_2 ($Nb_2 < Nb_1$) top-ranked images. Hence, even if the common features are neglected in the second step, this will not mislead the retrieval since they were considered in the first step. We confirmed experimentally, using the retrieval system described hereinbelow, the importance of processing queries with negative example in two steps.

[0048]   Figure 1 compares the curves precision-scope for the two techniques: negative example queries processed in two steps (the adopted technique) versus negative example queries processed in a unique step (in which both positive and negative examples are considered and all images in the database participate in retrieval). Precision is the average of relevance of retrieved images, and scope is the number of retrieved images. It is clear from Figure 1 that when queries containing negative example are considered in one step, the precision of retrieval decreases quickly with the number of retrieved images.

[0049]   Some special cases are important and merit to be mentioned to show that the proposed method functions as well. These cases emerge when all the discrimination features come from positive example only or from negative example only. Indeed, if the relevant features of positive example are strictly included in those of negative example and with common values, then applying the proposed principle leads, in the first step, to enhance the relevant features of positive example (which are the same as the common features) and to retain images looking like it. Then, in the second step, to enhance the rest of the negative example relevant features and to discard images near to it. On the other hand, if the relevant features of negative example are strictly included in those of positive example and with common values, then applying the proposed principle leads, in the first step, to enhance the relevant features positive

15

example (which include those of negative example) and to retain images looking like the positive example. Then, in the second step, to enhance only those features relevant to positive but not to negative example and to re-rank the images according to these features essentially.

[0050]   A last question that should be answered is: can the user compose a query using negative example only?

[0051]   First, we note that, for a given query, the number of non-relevant images is usually much higher than the number of relevant images. In other words, if we know what someone doesn't want, this doesn't inform us sufficiently about what the user wants. For example, if the user gives an image of a car as negative example without giving any positive example, then we cannot know whether the user is looking for images of buildings, animals, persons or other things. Nevertheless, we think that negative example can be used alone in some cases, for instance, to eliminate a subset from a database. Suppose that the database contains, in addition to images the user agrees with, other images that the user's culture doesn't tolerate, e.g. nudity images for some persons. In such a case, the user can first eliminate the undesired images by using some of them as negative example; then the user can navigate in, or retrieve from the rest of the database. Concerning the retrieval system, the negative-example-only query will be considered as a positive example query, i.e., the system first searches for images that resemble negative example. Then, when the resulting images (images that the user wants to discard) are retrieved, the system returns to the user the rest of the database rather these images.

[0052]   As explained hereinabove, the present invention generally aims at defining a retrieval scenario where the user can select positive example

16

images, negative example images, and their respective degrees of relevance. This allows first to reduce the heterogeneity of the dataset on the basis of the positive example, then to refine the results on the basis of the negative example.

[0053]   Via the interaction with the user, we globally aim to achieve two objectives. First, to be able to combine the query images together with their respective degrees of relevance in order to identify what the user is looking for; and to integrate this information in similarity measures. Second, to weight each feature and its components according to its relevance to the query and the discrimination power it can provide. Before we give details on our formulation of relevance feedback, we will begin by explaining the adopted image model and similarity measure. To represent images, we have adopted a hierarchical model like the one in [26] where each image, either in the query or in the database, is represented by a set of I features, each of which is a real vector of many components. This choice ensures a good modeling of both images and image features, and a reduction in the computation time [26]. The hierarchical two-level image model implies the need to choose a distance metric for each level. For feature level, we have chosen a generalized Euclidean distance function, as in [25]. If $\vec{x}_{i1}$ and $\vec{x}_{i2}$ are the ith feature vectors of the images x_1 and x_2 respectively, then the distance at this feature level is

$$D_i(\vec{x}_{i1}, \vec{x}_{i2}) = (\vec{x}_{i1} - \vec{x}_{i2})^T W_i(\vec{x}_{i1} - \vec{x}_{i2})$$

where W_i is a symmetric matrix that allows us to define the generalized ellipsoid distance D_i.

[0054]   The choice of this distance metric will allow us not only to weight each feature's component but also to transform the initial feature space

17

into a space that better models the user's needs and specificities [26]. The global distance between two images x_1 and x_2 is linear and is given in Equation

$$D(x_1, x_2) = \sum_{i=1}^{I} u_i (\vec{x}_{1i} - \vec{x}_{2i})^T W_i (\vec{x}_{1i} - \vec{x}_{2i})$$

(4)

where u_i is the global weight assigned to the ith feature.

[0055] Now that the image model and similarity metrics have been chosen, let us explain how to compute the parameters u_i and W_i which fulfill the objectives set in the previous section. Consider that the user constructs a query composed of N_1 positive example images and their respective relevance degrees π_n^1 for n = 1,...,N_1, as well as N_2 negative example images and their respective relevance degrees π_n^2 for n = 1,...,N_2. (It should be noted that π_n^2 is not the square of π_n; 2 is an index designating the negative example). In the first step, we consider only the positive example. We want to enhance each feature and its components according to their relevance to the positive example. This can be done by introducing the optimal parameters u_i and W_i which minimize $J_{positive}$, the global dispersion of positive example, given in Equation (5).

$$J_{positive} = \sum_{i=1}^{I} u_i \sum_{n=1}^{N_1} \pi_n^1 (\vec{x}_{ni}^1 - \vec{\bar{x}}_i^1)^T W_i (\vec{x}_{ni}^1 - \vec{\bar{x}}_i^1)$$

(5)

where $\vec{\bar{x}}_i^1$ is the weighted average of positive example (Figure 2), given by

18

$$\vec{x}_i^1 = \frac{\sum_{n=1}^{N_1} \pi_n^1 x_{ni}^1}{\sum_{n=1}^{N_1} \pi_n^1}$$

(6)

[0056] The problem formulation and the computation of optimal parameters are the same as in [26] excepted that the full set of query images is replaced by the positive example images only. The basic idea is to give more weight to features and feature components for which the positive example images are close to each other in the feature space. An informal justification is that if the variance of query images is high along a given axis, any value on this axis is apparently acceptable to the user, and therefore this axis should be given a low weight, and vice versa [25].

[0057] In the second step, we consider both positive and negative example images, and the refinement concerns the images retained in the first step. Let us first define J_{global}, the global dispersion of the query, including positive and negative example images:

$$J_{global} = \sum_{i=1}^{I} u_i \sum_{k=1}^{2} \sum_{n=1}^{N_k} \pi_n^k (\vec{x}_{ni}^k - \vec{q}_i)^T W_i (\vec{x}_{ni}^k - \vec{q}_i)$$

(7)

where k = 1 for positive example and k = 2 for negative example, and where $\vec{q}_i$, given in Equation (8), is the weighted average of all query images for the ith feature. See Figure 4.

$$\vec{q}_i = \frac{\sum_{k=1}^{2} \sum_{n=1}^{N_k} \pi_n^k \vec{x}_{ni}^k}{\sum_{k=1}^{2} \sum_{n=1}^{N_k} \pi_n^k}$$

(8)

[0058] Rui et al. proposed in [27] to allocate negative degrees of relevance to negative example images and to compute the parameters which

19

minimize the same expression of Equation (7). Let us analyze the consequences of such an approach. If we separate positive example from negative example in Equation (7), then we can write

$$J_{global} = \sum_{i=1}^{I} u_i \sum_{n=1}^{N_1} \pi_n^1 (\vec{x}_{ni}^1 - \vec{q}_i)^T W_i (\vec{x}_{ni}^1 - \vec{q}_i) + \sum_{i=1}^{I} u_i \sum_{n=1}^{N_2} \pi_n^2 (\vec{x}_{ni}^2 - \vec{q}_i)^T W_i (\vec{x}_{ni}^2 - \vec{q}_i)$$

[0059]   They choose $\pi_n^1 > 0$ for n = 1,...,N$_1$ and $\pi_n^2 < 0$ for n = 1,...,N$_2$. We thus obtain:

$$J_{global} = \sum_{i=1}^{I} u_i \sum_{n=1}^{N_1} \pi_n^1 (\vec{x}_{ni}^1 - \vec{q}_i)^T W_i (\vec{x}_{ni}^1 - \vec{q}_i) - \sum_{i=1}^{I} u_i \sum_{n=1}^{N_2} |\pi_n^2| (\vec{x}_{ni}^2 - \vec{q}_i)^T W_i (\vec{x}_{ni}^2 - \vec{q}_i)$$

(9)

[0060]   where $|\pi_n^2|$ designates the absolute value of π_n^2. Equation (9) shows that the global dispersion J$_{global}$ is nothing but the dispersion of positive example minus the dispersion of negative example. Hence, by minimizing the global dispersion, even if the authors move the global query average q (with which they compare their images) towards positive example and away from negative example, two problems emerge.

[0061]   First, minimizing the global dispersion will lead to minimize the dispersion of positive example, but with respect to the global query average q rather than the positive example average $\vec{x}_1$. This will not give an optimal minimization of the positive example dispersion; and hence, the relevant features of positive example will not be given enough importance.

[0062]   Second, minimizing the global dispersion will lead to maximize the dispersion of negative example. This implies that they neglect the

20

relevant features of negative example. Hence, their retrieval system will not be able to discard the undesired images. See Figure 3 for an illustration.

[0063]   Before describing our formulation of the problem in detail, let us recall the general objectives. By introducing the weights u_i and W_i, we want to give more importance to the relevant features of either positive or negative example which allow to distinguish well between them. In other words, via u_i and W_i we want to attribute weights to features and to transform our feature space into a new space in which positive example images are as close as possible, negative example images are as close as possible, and positive example is as far as possible from negative example. (See Figure 4). To translate our objectives into a mathematical formulation, we start by distinguishing positive example images from negative example images in the global dispersion formula of Equation (7). For each feature i, we recall the weighted average of positive example images $\bar{x}_i^1$ and we define the weighted average of negative example images $\bar{x}_i^2$ in Equations (10) and (11) respectively.

$$\bar{x}_i^1 = \frac{\sum_{n=1}^{N_1} \pi_n^1 x_{ni}^1}{\sum_{n=1}^{N_1} \pi_n^1}$$

(10)

$$\bar{x}_i^2 = \frac{\sum_{n=1}^{N_2} \pi_n^2 x_{ni}^2}{\sum_{n=1}^{N_2} \pi_n^2}$$

(11)

[0064]   By introducing $\bar{x}_i^1$ and $\bar{x}_i^2$ into Equation (7), we can rewrite it as follows:

21

$$J_{global} = \sum_{i=1}^{I} u_i \sum_{k=1}^{2} \sum_{n=1}^{N_k} \pi_n^k \left[(\vec{x}_{ni}^k - \vec{x}_i^k) + (\vec{x}_i^k - \vec{q}_i) \right]^T W_i \left[(\vec{x}_{ni}^k - \vec{x}_i^k) + (\vec{x}_i^k - \vec{q}_i) \right]$$

(12)

[0065]   Developing Equation (12) gives

$$J_{global} = \sum_{i=1}^{I} u_i \left[\left(\sum_{k=1}^{2} \sum_{n=1}^{N_k} \pi_n^k (\vec{x}_{ni}^k - \vec{x}_i^k)^T W_i (\vec{x}_{ni}^k - \vec{x}_i^k) \right) + \left(\sum_{k=1}^{2} \sum_{n=1}^{N_k} \pi_n^k (\vec{x}_{ni}^k - \vec{x}_i^k)^T W_i (\vec{x}_i^k - \vec{q}_i) \right) \right.$$
$$\left. + \left(\sum_{k=1}^{2} \sum_{n=1}^{N_k} \pi_n^k (\vec{x}_i^k - \vec{q}_i)^T W_i (\vec{x}_{ni}^k - \vec{x}_i^k) \right) + \left(\sum_{k=1}^{2} \sum_{n=1}^{N_k} \pi_n^k (\vec{x}_i^k - \vec{q}_i)^T W_i (\vec{x}_i^k - \vec{q}_i) \right) \right]$$

(13)

[0066]   It can easily be shown that the second and third parts of Equation (13) are zero. For example, the second part

$$\sum_{k=1}^{2} \sum_{n=1}^{N_k} \pi_n^k (\vec{x}_{ni}^k - \vec{x}_i^k)^T W_i (\vec{x}_i^k - \vec{q}_i) = \sum_{k=1}^{2} \left[\left(\sum_{n=1}^{N_k} \pi_n^k (\vec{x}_{ni}^k - \vec{x}_i^k)^T \right) W_i (\vec{x}_i^k - \vec{q}_i) \right]$$

$$= \sum_{k=1}^{2} \left[\left(\left(\sum_{n=1}^{N_k} \pi_n^k \vec{x}_{ni}^k \right) - \left(\sum_{n=1}^{N_k} \pi_n^k \right) \vec{x}_i^k \right)^T W_i (\vec{x}_i^k - \vec{q}_i) \right] = 0$$

since, according to Equations (10) and (11),

$$\sum_{n=1}^{N_k} \pi_n^k x_{ni}^k - \left(\sum_{n=1}^{N_k} \pi_n^k \right) \vec{x}_i^k = 0$$

[0067]   Thus, Equation (13) can be written as follows:

$$J_{global} = \left[\sum_{i=1}^{I} u_i \sum_{k=1}^{2} \sum_{n=1}^{N_k} \pi_n^k (\vec{x}_{ni}^k - \vec{x}_i^k)^T W_i (\vec{x}_{ni}^k - \vec{x}_i^k) \right] + \left[\sum_{i=1}^{I} u_i \sum_{k=1}^{2} \sum_{n=1}^{N_k} \pi_n^k (\vec{x}_i^k - \vec{q}_i)^T W_i (\vec{x}_i^k - \vec{q}_i) \right] = A + R$$

(14)

[0068]   The first term A expresses the positive example internal dispersion, i.e., how close positive example images are to each other, added to the negative example internal dispersion, i.e., how close negative example

22

images are to each other. The second term R expresses the distance between the two sets, i.e., how far positive example is from negative example.

[0069]   By distinguishing the intra dispersion A from the inter dispersion R, it is now clearer how we can formulate our objectives in a mathematical problem. In fact, we want to compute the model parameters, namely u_i and W_i, which minimize the intra dispersion A and maximize the inter dispersion R. Several combinations of A and R are possible. We have chosen to compute the parameters which minimize the ratio $\frac{A}{R}$, assuming that $R \neq 0$. In the case of $R = 0$, the positive example and the negative example are not distinguishable and the query is ambiguous. In such case, the query is rejected and the user is asked to formulate a new one. Furthermore, to avoid numerical stability problems, we introduce the following two constraints: $\sum_{i=1}^{I} \frac{1}{u_i} = 1$ and det(W_i)=1 for all i=1,...,I. By using Lagrange multipliers, the optimal parameters u_i and W_i must minimize the quantity L given in Equation (15).

$$L = \frac{A}{R} - \lambda\Big(\sum_{i=1}^{I} \frac{1}{u_i} - 1\Big) - \sum_{i=1}^{I} \lambda_i(det(W_i) - 1)$$

(15)

where

$$A = \sum_{i=1}^{I} u_i \sum_{k=1}^{2} \sum_{n=1}^{N_k} \pi_n^k (\vec{x}_{ni}^k - \vec{\bar{x}}_i^k)^T W_i (\vec{x}_{ni}^k - \vec{\bar{x}}_i^k)$$

(16)

and

$$R = \sum_{i=1}^{I} u_i \sum_{k=1}^{2} \bar{\pi}^k (\vec{\bar{x}}_i^k - \vec{q}_i)^T W_i (\vec{\bar{x}}_i^k - \vec{q}_i)$$

(17)

23

$\tilde{\pi}^1$ denotes the sum of positive example relevance degrees, i.e., $\tilde{\pi}^1 = \sum_{n=1}^{N_1} \pi_n^1$

and $\tilde{\pi}^2$ denotes the sum of negative example relevance degrees, i.e.,

$\tilde{\pi}^2 = \sum_{n=1}^{N_2} \pi_n^2$.

[0070]   The optimization problem in order to obtain the optimal parameters u_i and W_i will now be resolved.

[0071]   Before starting, we note that the relative importance of positive and negative examples must be chosen, i.e., $\tilde{\pi}^1$ with respect to $\tilde{\pi}^2$. Some image retrieval systems [7] adopt the values used by certain text retrieval systems which are 0.65 for positive example and 0.35 for negative example. Other systems such as the one in [20] assume that positive example and negative example have the same importance. We have adopted for the latter choice because it allows some simplifications in the derivation of the problem. Furthermore, we normalize all the user-given relevance degrees so that $\tilde{\pi}^1 + \tilde{\pi}^2 = 1$.

[0072]   To obtain the optimal solution for W_i, we take the partial derivative of L with respect to $w_{i_{rs}}$ for r,s=1,...,H_i, where H_i is the dimension of the ith feature and $w_{i_{rs}}$ is the rsth element of W_i, i.e., $W_i = [w_{i_{rs}}]$. We have

$$\frac{\partial L}{\partial w_{i_{rs}}} = \frac{R \frac{\partial A}{\partial w_{i_{rs}}} - A \frac{\partial R}{\partial w_{i_{rs}}}}{R^2} - \lambda_i \frac{\partial det(W_i)}{\partial w_{i_{rs}}} \qquad (18)$$

where

$$\frac{\partial A}{\partial w_{i_{rs}}} = u_i \sum_{k=1}^{2} \sum_{n=1}^{N_k} \pi_n^k (x_{ni_r}^k - \bar{x}_{i_r}^k)(x_{ni_s}^k - \bar{x}_{i_s}^k) \qquad (19)$$

24

and

$$\frac{\partial R}{\partial w_{ir_s}} = u_i \sum_{k=1}^{2} \bar{\pi}^k (\bar{x}_{i_r}^k - q_{i_r})(\bar{x}_{i_s}^k - q_{i_s})$$

(20)

[0073]   Before   computing $\dfrac{\partial L}{\partial w_{i_{rs}}}$,   we   note   that

$\det(W_i) = \sum_{r=1}^{H_i}(-1)^{r+s} w_{i_{rs}} \det(W_{i_{rs}})$, where $\det(W_{i_{rs}})$ is the rs[th] minor of W_i obtained

by eliminating the rth row and the sth column of $\det(W_i)$. Hence,

$$\frac{\partial \det(W_i)}{\partial w_{ir_s}} = (-1)^{r+s} \det(W_{ir_s})$$

(21)

By substituting Equations (19), (20) and (21) in (18), we obtain

$$\frac{\partial L}{\partial w_{i_{rs}}} = 0 \Leftrightarrow$$

$$R\left[u_i \sum_{k=1}^{2}\sum_{n=1}^{N_k} \pi_n^k(x_{ni_r}^k - \bar{x}_{i_r}^k)(x_{ni_s}^k - \bar{x}_{i_s}^k)\right] - A\left[u_i \sum_{k=1}^{2}\pi^k(\bar{x}_{i_r}^k - q_{i_r})(\bar{x}_{i_s}^k - q_{i_s})\right] - R^2\lambda_i(-1)^{r+s}\det(W_{i_{rs}}) = 0 \Leftrightarrow$$

$$\det(W_{i_{rs}}) = \frac{u_i}{(-1)^{r+s}\lambda_i R^2}\left[R\sum_{k=1}^{2}\sum_{n=1}^{N_k}\pi_n^k(x_{ni_r}^k - \bar{x}_{i_r}^k)(x_{ni_s}^k - \bar{x}_{i_s}^k) - A\sum_{k=1}^{2}\pi^k(\bar{x}_{i_r}^k - q_{i_r})(\bar{x}_{i_s}^k - q_{i_s})\right]$$

(22)

[0074]   Now consider the matrix $W_i^{-1} = [w_{i_{rs}}^{-1}]$, the inverse matrix of W_i

(provided that W_i is invertible). To obtain the value of each component $w_{i_{rs}}^{-1}$, we

use the determinant method for matrix inversion to obtain

$$w_{ir_s}^{-1} = \frac{(-1)^{r+s}\det(W_{ir_s})}{\det(W_i)}$$

25

Knowing that det(W$_l$)=1, we find

$$w_{i_{rs}}^{-1} = (-1)^{r+s} det(W_{i_{rs}})$$

(23)

[0075]   In Equation (23), we replace det($W_{i_{rs}}$) by its value from Equation (22) to obtain

$$w_{i_{rs}}^{-1} = \frac{1}{\gamma}\left[R\sum_{k=1}^{2}\sum_{n=1}^{N_k} \pi_n^k (x_{ni_r}^k - \bar{x}_{i_r}^k)(x_{ni_s}^k - \bar{x}_{i_s}^k) - A\sum_{k=1}^{2} \bar{\pi}^k (\bar{x}_{i_r}^k - q_{i_r})(\bar{x}_{i_s}^k - q_{i_s}) \right]$$

(24)

where

$$\gamma = \frac{\lambda_i R^2}{u_i}$$

[0076]   Equation (24) can also be written in matrix form as

$$W_i^{-1} = \frac{1}{\gamma} C_i$$

(25)

where C$_i$ is the matrix [$c_{i_{rs}}$] such that

$$c_{i_{rs}} = R\sum_{k=1}^{2}\sum_{n=1}^{N_k} \pi_n^k (x_{ni_r}^k - \bar{x}_{i_r}^k)(x_{ni_s}^k - \bar{x}_{i_s}^k) - A\sum_{k=1}^{2} \bar{\pi}^k (\bar{x}_{i_r}^k - q_{i_r})(\bar{x}_{i_s}^k - q_{i_s})$$

(26)

[0077]   Now we will compute the value of γ independently from λ which is an unknown parameter. We can write Equation (25) as follows:

$$W_i^{-1} = \frac{1}{\gamma} C_i \Leftrightarrow C_i = \gamma W_i^{-1} \Rightarrow det(C_i) = \gamma^{H_i} det(W_i^{-1})$$

26

but since $\det(W_i^{-1}) = 1$, then $\gamma = (\det(C_i))^{\frac{1}{H_i}} C_i^{-1}$. Finally, the optimal solution for W_i is given by Equation (27)

$$W_i = \gamma C_i^{-1} = (det(C_i))^{\frac{1}{H_i}} C_i^{-1} \tag{27}$$

where the components of C_i are given by Equation (26).

[0078]   Let us analyze, roughly, the effect of the dispersion of positive and negative examples on the components of W_i. First, we can write Equation (26) in a matrix form, as follows:

$$C_i = RCova_i - ACovr_i \tag{28}$$

where $Cova_i$ is the sum of intra covariance matrices for the ith feature, i.e., $Cova_i = [\mathrm{cov}\, a_{i_{rs}}]$ such that

$$cova_{i_{rs}} = \sum_{k=1}^{2} \sum_{n=1}^{N_k} \pi_n^k (x_{ni_r}^k - \bar{x}_{i_r}^k)(x_{ni_s}^k - \bar{x}_{i_s}^k)$$

and $Covr_i$ is the inter covariance matrix for the ith feature, i.e., $Covr_i = [\mathrm{cov}\, r_{i_{rs}}]$ such that

$$covr_{i_{rs}} = \sum_{k=1}^{2} \tilde{\pi}^k (\bar{x}_{i_r}^k - q_{i_r})(\bar{x}_{i_s}^k - q_{i_s})$$

[0079]   Now, consider Equation (28), where we set the values of A and R since they concern all the features. If the intra dispersion is high relative to the inter dispersion, and hence the elements of $Cova_i$ are important relative to the elements of $Covr_i$ then, according to Equation (28), the values of the

27

components of C_I will be important. But since $W_i = \gamma C_i^{-1}$ (Equation(27)), it follows that the values of w_{i_n} will be small; and consequently, the ith feature's components will be given low weights. On the other hand, if the intra dispersion is low relative to the inter dispersion for the ith feature, by a similar line of reasoning, we see that this feature's components will be given high weights. This behavior of W_I fulfills our objective of enhancing discriminant features against other ones.

[0080] To obtain the optimal solution for u_i, we take the partial derivative of L with respect to u_i.

$$\frac{\partial L}{\partial u_i} = \frac{R\frac{\partial A}{\partial u_i} - A\frac{\partial R}{\partial u_i}}{R^2} + \frac{\lambda}{u_i^2} \tag{29}$$

where

$$\frac{\partial A}{\partial u_i} = \sum_{k=1}^{2} \sum_{n=1}^{N_k} \pi_n^k (\vec{x}_{ni}^k - \vec{\bar{x}}_i^k)^T W_i (\vec{x}_{ni}^k - \vec{\bar{x}}_i^k) \tag{30}$$

and

$$\frac{\partial R}{\partial u_i} = \sum_{k=1}^{2} \tilde{\pi}^k (\vec{\bar{x}}_i^k - \vec{q}_i)^T W_i (\vec{\bar{x}}_i^k - \vec{q}_i) \tag{31}$$

[0081] By substituting Equations (30) and (31) in (29), we obtain

$$\frac{\partial L}{\partial u_i} = 0 \Leftrightarrow R\left[\sum_{k=1}^{2}\sum_{n=1}^{N_k} \pi_n^k(\vec{x}_{ni}^k - \vec{\bar{x}}_i^k)^T W_i(\vec{x}_{ni}^k - \vec{\bar{x}}_i^k)\right] - A\left[\sum_{k=1}^{2} \tilde{\pi}^k(\vec{\bar{x}}_i^k - \vec{q}_i)^T W_i(\vec{\bar{x}}_i^k - \vec{q}_i)\right] + \frac{\lambda R^2}{u_i^2} = 0 \tag{32}$$

[0082] We multiply both sides of Equation (32) by u_i, to obtain:

28

$$u_i f_i + \frac{\lambda R^2}{u_i} = 0$$

(33)

where

$$f_i = R\Big[\sum_{k=1}^{2}\sum_{n=1}^{N_k}\pi_n^k(\bar{x}_{ni}^k - \bar{x}_i^k)^T W_i(\bar{x}_{ni}^k - \bar{x}_i^k)\Big] - A\Big[\sum_{k=1}^{2}\bar{\pi}^k(\bar{x}_i^k - \bar{q}_i)^T W_i(\bar{x}_i^k - \bar{q}_i)\Big]$$

(34)

[0083]   Now, to get rid of the unknown parameter λ, we will try to find a relation, independent of λ, between u_i and any u_j. First λ can be computed directly from Equation (33) as follows:

$$\lambda = -\frac{f_i u_i^2}{R^2}\ \forall i$$

(35)

[0084]   Second, taking the sum on i of Equation (33) gives $\sum_{i=1}^{I}u_j f_j + \lambda R^2 \sum_{j=1}^{I}\frac{1}{u_j} = 0$, but since $\sum_{i=1}^{I}\frac{1}{u_i} = 1$, then $\sum_{i=1}^{I}u_j f_j + \lambda R^2 = 0$. It follows that

$$\lambda = \frac{-\sum_{i=j}^{I}u_j f_j}{R^2}$$

(36)

[0085]   Equations (35) and (36) imply that for every feature i

$$f_i u_i^2 = \sum_{j=1}^{I}u_j f_j$$

(37)

[0086]   It follows from Equation (37) that $f_1 u_1^2 = f_2 u_2^2 = ... = f_i u_i^2 = f_I u_I^2$.

[0087]   Hence,

29

$$u_j = u_i \sqrt{\frac{f_i}{f_j}} \quad \forall j$$

(38)

[0088] Finally, to obtain the optimal solution of u_i, we replace u_j in Equation (37) by its value from Equation (38). We find the following:

$$f_i u_i^2 = \sum_{j=1}^{I} \left(u_i \sqrt{\frac{f_i}{f_j}} f_j \right) \Leftrightarrow f_i u_i = \sum_{j=1}^{I} \sqrt{f_i f_j}$$

$$\Leftrightarrow u_i = \frac{\sum_{j=1}^{I} \sqrt{f_j}}{\sqrt{f_i}}$$

(39)

[0089] The optimal solution for u_i is given by Equation (39), where f_i is defined by Equation (34).

[0090] Let us analyze, roughly, the influence of the dispersion of positive and negative examples on the value of each u_i. First, we can write f_i in Equation (34) as

$$f_i = RFa_i - AFr_i$$

(40)

where

$$Fa_i = \sum_{k=1}^{2} \sum_{n=1}^{N_k} \pi_n^k (\vec{x}_{ni}^k - \vec{\bar{x}}_i^k)^T W_i (\vec{x}_{ni}^k - \vec{\bar{x}}_i^k)$$

(41)

and

$$Fr_i = \sum_{k=1}^{2} \tilde{\pi}^k (\vec{\bar{x}}_i^k - \vec{q}_i)^T W_i (\vec{\bar{x}}_i^k - \vec{q}_i)$$

(42)

30

[0091]   We assume that A and R have constant values since they depend on all the features. If, for the ith feature, the intra dispersion is high relative to the inter dispersion, then the quantity Fa_i will gain in importance relative to the quantity Fr_i. According to Equation (40), this will increase the value of f_i. Moreover, Equation (39) shows that when f_i increases, u_i decreases; and hence, the ith feature will be given a low weight. Conversely, if, for the ith feature, the intra dispersion is low relative to the inter dispersion, then, by a similar line of reasoning, we find that the ith feature will be given a high weight. Therefore, the optimal value that we found for u_i fulfills our objective of enhancing the relevant discriminant features against others.

[0092]   The input to the algorithm consists of positive example images, negative example images and their respective relevance degrees. The output consists of the optimal parameters W_l and u_l. These parameters are computed according to Equations (27) and (39), respectively. The computation of these parameters requires the computation of $\bar{x}_{i.}^{1}$, $\bar{x}_{i}^{2}$, $\bar{q}_{i}$, f_i, A and R according to Equations (10), (11), (8), (34), (16) and (17), respectively. Our algorithm is iterative because the computation of W_l and u_i depends on A and R, and the computation of A and R depends on W_l and u_i. We use the fixed point method to perform the computation of W_l and u_i. An initialization step is required, in which we adopt the following values:

[0093]   - We initialize W_l with the diagonal matrix

31

$$
\begin{pmatrix}
\frac{1}{\sigma_{i_1}} & \cdot & \cdot & 0 \\
\cdot & \cdot & & \cdot \\
\cdot & & \cdot & \cdot \\
0 & \cdot & \cdot & \frac{1}{\sigma_{i_{H_i}}}
\end{pmatrix}
$$

where

$$
\sigma_{ir} = \sqrt{\sum_{k=1}^{2}\sum_{n=1}^{N_k} \pi_n^k (x_{ni_r}^k - q_{i_r})^2}
$$

is the standard deviation of the rth component of the ith feature computed for the full set of query images.

[0094] - We initialize the parameter u_i with a kind of dispersion given by

$$
u_i = \frac{\sum_{j=1}^{I} \sqrt{f_j}}{\sqrt{f_i}}
$$

where

$$
f_i = \frac{\sum_{k=1}^{2}\sum_{n=1}^{N_k} \pi_n^k (\vec{x}_{ni}^k - \vec{x}_i^k)^T W_i (\vec{x}_{ni}^k - \vec{x}_i^k)}{\sum_{k=1}^{2} \tilde{\pi}^k (\vec{x}_i^k - \vec{q}_i)^T W_i (\vec{x}_i^k - \vec{q}_i)}
$$

[0095] The computation of W_i requires the inversion of the matrix C_i. However, in the case of $(N_1+N_2)<H_i$, C_i is not invertible. In [25], the authors suggest proceeding by singular value decomposition (SVD) to obtain the pseudo inverse matrix. This solution doesn't give a satisfactory result,

32

especially when $(N_1+N_2)$is far less than H_l [26]. The authors of [26] propose, in the case of a singular matrix, to replace W_l by a diagonal matrix whose elements are the inverse of the standard deviation, i.e., $w_{i_{rs}} = \dfrac{1}{\sigma_{i_r}}$ if r = s and $w_{i_{rs}} = 0$ elsewhere. In our case, we propose to replace W_l by a diagonal matrix whose elements are the inverse of the diagonal elements of the matrix C_l, i.e.,

$$W_i = \begin{pmatrix} w_{i_{11}} & \cdot & \cdot & 0 \\ \cdot & \cdot & & \cdot \\ \cdot & & \cdot & \cdot \\ 0 & \cdot & \cdot & w_{i_{H_i H_i}} \end{pmatrix}$$

where $w_{i_{ss}} = \dfrac{1}{c_{i_{ss}}}$ and $c_{i_{ss}}$ can be obtained by setting r = s in Equation (26).

Example of application

[0096]   The relevance feedback model was validated in the context of image retrieval. To do so, we implemented a content-based image retrieval system in which we present the user with a graphical interface displaying nine sample images related to different subjects and emphasizing different features (see Figure 5). The user can choose more images from the database before formulating the query. To add an image to the query, the user clicks on its "Select" button. The system displays a dialog box allowing the user to specify a degree of relevance (see Figure 6).

[0097]   The possible relevance degrees are

- Very similar: corresponds to the relevance value 2 for a positive example image.

- Similar: corresponds to the relevance value 1 for a positive example image.

- Doesn't matter: the image will not participate in the query.

- Different: corresponds to the relevance value 1 for a negative example image.

- Very different: corresponds to the relevance value 2 for a negative example image.

[0098] Once the query formulation is completed, the user can launch the retrieval process by clicking on the "Begin search" button. The system computes the optimal parameters and performs comparison in three ways depending on the constitution of the query. If the query contains only positive example images, then the optimal parameters are those which minimize Equation (5), and the database images are ranked in increasing order according to their distance from the positive example average. This distance is given by Equation (43). We return the top-ranked images to the user.

$$D(x_n) = \sum_{i=1}^{I} u_i (\vec{x}_{ni} - \vec{x}_i^1)^T W_i (\vec{x}_{ni} - \vec{x}_i^1) \tag{43}$$

[0099] If the query contains only negative example images, then the system proceeds initially by a similar procedure, but considering the negative example rather than the positive example. This means that the system computes the ideal parameters which minimize the dispersion of negative example images, ranks the images in increasing order according to their distance from the negative example average, then returns to the user the last-

34

ranked images. If the query contains both positive and negative examples, then the system performs the two steps of retrieval. The parameter computation and the distance function used in the first step are the same as in the case of a positive-example-only query. The database images are ranked in increasing order and only the Nb_1 first ones are retained for the second step. In the second step, the system computes the parameters which minimize Equation (15), then the retained images are ranked according to their closeness to the positive example and their farness from the negative example. The comparison function is given by Equation (44). Finally, the system returns the Nb_2 top-ranked images to the user.

$$D(x_n) = \sum_{i=1}^{I} u_i(\vec{x}_{ni} - \vec{x}_i^1)^T W_i(\vec{x}_{ni} - \vec{x}_i^1) - \sum_{i=1}^{I} u_i(\vec{x}_{ni} - \vec{x}_i^2)^T W_i(\vec{x}_{ni} - \vec{x}_i^2)$$

(44)

[00100]   Each image, either in the database or in the query, is represented by a set of 27 feature vectors, computed as follows: First, we map every pixel in the image to a point in the 3-D HSI space (Figure 7). This operation consists of computing, for every triple [H,S,I], the number of pixels having the values Hue = H, Saturation = S and Intensity = I. We obtain a 3-D color histogram that we can't use, as it takes up a lot of space and most of its values are zeros. For example, if we have an image with HSI values ranging between 0 and 255, our histogram will contain 256^3 cells, most of which don't correspond to any pixel.

[00101]   To reduce the histogram's size, many solutions are possible, such as the spatial repartition of the points of the 3-D histogram, taking into account their respective occurrence frequency, i.e., the number of pixels corresponding to each point in the histogram. However, as the main aim of our work is not finding the best visual features, but rather developing a relevance

35

feedback technique, we have adopted a simple solution which consists of partitioning the space by subdividing the axes H, S and I into three equal intervals each. This gives us 3^3 = 27 subspaces, as shown in Figure 7. Each subspace will constitute a feature, and we compute its corresponding vector as follows. The subspace is subdivided into 2^3 = 8 sub-subspaces. We compute the sum of the elements of each sub-subspace and we put the result in the corresponding cell of the feature vector (Figure 7). We note that our feature estimation method is somewhat similar to the quantizer function used in [15].

Experimental results and performance evaluation

Example of refinement with negative example

[00102]   Tests were performed on 10.000 images from The Pennsylvania State University [16, 13]. This database contains images related to different subjects, emphasizing different features, and taken under different illumination conditions. For each image, the set of features is computed as explained above. We performed many tests for retrieval and refinement. Even when positive and negative examples are not readily distinguishable, a system according to the present invention succeeded in identifying discrimination features and sorting the resulting images according to these features. Figure 8 gives an example of retrieval with positive example only. Figure 9 gives and example of retrieval with positive and negative examples.

[00103]   In the first example, two images participated in the query as positive example. Both of these images contain a green tree under the blue sky (5095.ppm and 5118.ppm). Figure 8 shows the top nine returned images. We notice that the two query images are returned in the top positions. There are also some other images containing trees under the sky, but we notice the presence of noise consisting of three images of a brown bird on a green tree

under the blue sky (5523.ppm, 5522.ppm, 5521.ppm). At the same time, there have been miss, because the database contains other images of trees under the sky that have not been retrieved. In the second example, we try to refine the results of the first example. Hence, we use the same images (5095.ppm and 5118.ppm) as positive example, while an image of a bird on a tree under the sky is chosen as negative example (image 5521.ppm of Figure 8). Figure 9 shows that images of birds are discarded (the noise reduced) and that more images of trees under the sky are retrieved (the miss decreased).

Performance evaluation

[00104]   In order to validate the proposed relevance feedback technique, we performed a performance evaluation of our retrieval system. The evaluation was based on comparison between the use of positive example only and the use of both positive and negative examples. To perform any evaluation in the context of image retrieval, two main issues emerge: the acquisition of ground truth and the definition of performance criteria. For ground truth, we use human subjects: three persons participated in the all the experiences that we will describe below. Concerning performance criteria, the most used are Precision Pr and Recall Re [14]. In their simplest definition, precision is the proportion of retrieved images that are relevant, i.e., number of retrieved images that are relevant on the number of all retrieved images; and recall is the proportion of relevant images that are retrieved, i.e., number of relevant images that are retrieved on the number of all relevant images in the database. Some authors drew up the precision-recall curve Pr=f(Re) [14]; however, it has been observed, that this measure is less meaningful in the context of image retrieval since recall is consistently low [26] [11]. Furthermore, we think that it is often difficult to compute recall, especially when the size of the image database is big; because this requires to know, for each query, the number of relevant images in a the whole database. Another problem with recall, is that it depends

37

strongly on the choice of the number of images to return to the user. If the number of relevant images in the database is bigger than the number of images returned to the user, then the recall will be penalized. We adopt a more expressive curve which is the precision-scope curve Pr=f(Sc) [11]. Scope Sc is the number of images returned to the user, and hence the curve Pr=f(Sc) depicts the precision for different values of the number of images returned to the user.

[00105]   We carried out two experiences, each of which tries to measure a given aspect of our model. The first experience aims to measure the improvement, with negative example, in the relevance of retrieved images. The second experience aims to measure the improvement, with negative example, in the number of iterations needed to locate a given category of images.

First experience

[00106]   As mentioned above, the goal of the first experience is to measure the contribution of negative example in the   improvement of the relevance of retrieved images. Each human subject participating in the experience was asked to formulate a query using only positive example and to give a goodness score to each retrieved image, then to refine the results using negative example and to give a goodness score to each retrieved image. The possible scores are 2 if the image is good, 1 if the image is acceptable, and 0 if the image is bad. Each subject repeated the experience five times by specifying a new query each time. We computed precision as follows: Pr = the sum of degrees of relevance for retrieved images / the number of retrieved images. Figure 10 compares the curves Pr=f(Sc) in the two cases: retrieval with positive example and refinement with negative example.

38

[00107]   We find that in average, when we introduce negative example, the improvement in precision is about 20 %. In fact, the improvement varies from one query to another, because it depends on other factors such as the choice of a meaningful negative example and the constitution of the database. If, for a given query, the database contains a little number of relevant images, most of which have been retrieved in the first step, then the introduction of negative example or any other technique will not be able to bring any notable improvement.

Second experience

[00108]   The second experience attempts to measure the improvement in the number of refinement iterations needed to locate a given category of images, as well as the role of negative example in resolving the page zero problem (finding a good image to initiate the retrieval). We showed to each of our human subjects a set of images that are relatively similar to each other with respect to the color. None of the showed images appear in the set of images the subjects can use to formulate the initial query. Each subject is asked to locate at least one of the showed images using only positive example, and to count the number of iterations; then to restart the experience but using both positive and negative examples, and to count the number of iterations. This experience was repeated four times and the results are given in Figure 11. S1, S2 and S3 designate respectively the three human subjects who participated in the experiments. PE means positive example and NE means negative example. Each entry in the table gives the number of iterations needed to locate the searched images.

[00109]   Our first observation is that when they used both positive and negative examples, the subjects succeeded in all the experiences; however, when they used only positive example, some of them failed in certain

39

experiences to locate any sought image. In Experience 2.2 and Experience 2.4, at least one subject was unable to locate any sought image using positive example only. This is because, in a given iteration, all the retrieved images fall into an undesired category, and the formulation of the next-iteration query using any of these images leads to retrieve images belonging to the same category. The user can loop indefinitely, but will not be able to escape this situation by using positive example only. The second observation is that the use of negative example reduces appreciably the number of iterations. If we compute the average number of iterations among the successful experiences (2.1 and 2.3), we find 5.83 when only positive example is used, and 2.33 when both positive and negative examples are used. This experience shows clearly the role of negative example in mitigating the page zero problem. Indeed, after having obtaining at least one of the sought images, the user can use it to formulate a new query, and hence to retrieve more sought images.

[00110]   When performing content-based image retrieval, it is very important to take into account the user's needs and specificities, which can be identified via relevance feedback. However, the use of positive example only isn't always sufficient to determine what the user is looking for. This can be seen especially when all the candidate images to participate in the query appear in an inappropriate context or contain, in addition to the features the user is looking for, features or objects that the user doesn't want to retrieve. Motivated by the importance of negative example, in this paper we studied its relevance in content-based image retrieval. We gave details and justifications on how it can be combined with positive example to perform feature weighting, and then explained how this can be applied to the retrieval process. This led us to propose a new model for positive and negative feedback. We validated our model by testing it on a heterogeneous database and performing some performance evaluations.

40

[00111]   It is to be noted that the present model is not limited to image retrieval but can be adapted and applied to any retrieval process with relevance feedback.

[00112]   Although the present invention has been described hereinabove by way of preferred embodiments thereof, it can be modified, without departing from the spirit and nature of the subject invention.

41

References

[00113]   [1] A. Trouvé and Y. Yu. Metric Similarities Learning Through Examples: An application to Shape Retrieval. In 3rd EMMCVPR International Workshop, pages 50--62, Sophia-Antipolis, 2001.

[00114]   [2] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(12):1349–1380, 2000.

[00115]   [3] B. Bhanu, J. Peng, and S. Qing. Learning Feature Relevance and Similarity Metrics in Image Databases. In Workshop in Content-based Access to Image and Video Libraries, pages 14--18, 1998.

[00116]   [4] C. Carson, M. Thomas, S. Belongie, J.M. Hellerstein, and J. Malik. Blobworld: A system for region-based image indexing and retrieval. In International Conference on Visual Information Systems, pages 509--516, Amsterdam, 1999.

[00117]   [5] C. Meilhac, and C. Nastar. Relevance Feedback and Category Search in Image Databases. In IEEE International Conference on Multimedia Computing and Systems, pages 512--517, Florence, Italy, 1999.

[00118]   [6] C. Nastar, M. Mitschke, and C. Meilhac. Efficient Query Refinement for Image Retrieval. In IEEE Conference on Computer Vision and Pattern Recognition, pages 547–552, Santa Barbara, 1998.

[00119]   [7] H. Müller, W. Müller, D.M. Squire, S. Marchand-Maillet, and T. Pun. Strategies for positive and negative relevance feedback in image retrieval. Technical Report N° 00.01, Computer Vision Group, Computing Center, University of Geneva, 2000.

[00120]   [8] H.J. Zhang, Z. Chen, W.Y. Liu and M. Li. Relevance Feedback in Content-Based Image Search. Invited Keynote Speech in the 12th International Conference on New Information Technology (NIT). Beijing, May 2001.

[00121]   [9] I.J. Cox, M.L. Miller, S.M. Omohundro and P.N. Yianilos. PicHunter: Bayesian Relevance Feedback for Image Retrieval. In International Conference on Pattern Recognition, pages 361--369, Vienna, Austria, 1996.

[00122]   [10] J. Han, Y. Cai, and N. Cercone. Knowledge discovery in databases: An attribute-oriented approach. In 18th International Conference on Very Large Data Bases, pages 547--559, Vancouver, 1992.

[00123]   [11] J. Huang, S. Kumar, M. Mitra, W.-J. Zhu and R. Zabih. Image Indexing using Color Correlogram. In IEEE Conference on Computer Vision and Pattern Recognition, 1997.

[00124]   [12] J. J. Rocchio. Relevance Feedback in information retrieval. In SMART Retrieval System, Experiments in Automatic Document Processing, pages 323--323, New Jersey, 1971.

43

[00125]   [13] J. Li, J.Z. Wang and G. Wiederhold. IRM: Integrated region matching for image retrieval. In 2000 ACM Multimedia Conference, pages 147--156, San Jose, USA, 2000.

[00126]   [14] John R. Smith. Image Retrieval Evaluation. In IEEE Workshop on Content-based Access of Image and Video Libraries, 1998.

[00127]   [15] J.R. Smith and S.F. Chang. Tools and techniques for color image retrieval. In Storage & Retrieval for Image and Video Databases IV, vol. 2670 of IS&T/SPIE Proceedings, pages 426--437, San Jose, USA, 1996.

[00128]   [16] J.Z. Wang, J. Li and G. Wiederhold. SIMPLIcity: Semantics-sensitive Integrated Matching for Picture Libraries. IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(9):947--963, 2001.

[00129]   [17] L. Taycher, M. La cascia, S. Sclaroff. Image Digestion and Relevance Feedback in the ImageRover WWW Search Engine. In Second International Conference on Visual Information Systems, pages 85--91, San Diego, 1997.

[00130]   [18] M. Flickner, H. Sawhney, W. Niblack, J. Sahley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. Query by image and video content: The QBIC system. IEEE Computer Magazine, 28:23--32, 1995.

44

[00131] [19] M.L. Kherfi, D. Ziou and A. Bernardi. Web Image Search Engines: A Survey. Technical Report N° 276, Université de Sherbrooke, Canada, December 2001.

[00132] [20] N. Vasconcelos and A. Lippman. Learning from User Feedback in Image Retrieval Systems. In Neural Information Processing Systems 12, Denver, Colorado, 1999.

[00133] [21] N.J. Belkin, J. Perez-Carballo, C. Cool, S. Lin, S.Y. Park, S.Y. Rieh, P. Savage, C. Sikora, H. Xie, and J. Allan. Rutgers' TREC-6 interactive track experience. In 6th Text Retrieval Conference, pages 597–610, Gaitherburg, USA, 1998.

[00134] [22] R.S. Michalski. A Theory and Methodology of Inductive Learning. In Morgan Kaufmann, editor, Michalski et. al. (eds.), Machine Learning: An Artificial Intelligence Approach, vol. 1, pages 83–134, 1983.

[00135] [23] R.W. Picard, T.P. Minka, and M. Szummer. Modeling user subjectivity in image libraries. Technical Report N° 382, MIT Media Lab Perceptual Computing, 1996.

[00136] [24] T.P. Minka and R.W. Picard. Interactive learning using a 'society of models'. In IEEE Conference on Computer Vision and Pattern Recognition, pages 447–452, San Francisco, 1996.

[00137] [25] Y. Ishikawa, R. Subramanya, and C. Faloutsos. Mindreader: Query databases through multiple examples. In 24th International Conference on Very Large Data Bases, pages 433–438, New York, 1998.

45

[00138]   [26] Y. Rui and T.S. Huang. Optimizing Learning in Image Retrieval. In IEEE international Conference On Computer Vision and Pattern Recognition, Hilton Head, Sc, USA, 2000.

[00139]   [27] Y. Rui et al. Efficient indexing, browsing and retrieval of image/video content. PhD thesis, Department of Computer Science, University of Illinois at Urbana-Champaign, 1999.

[00140]   [28] Y. Rui, {T.S.} Huang, and S. Mehrotra. Content-based image retrieval with relevance feedback in MARS. In IEEE International Conference on Image Processing, pages 815--818, Santa Barbara, California, 1997.

[00141]   [29] Y. Rui, T.S. Huang, and S.F. Chang. Image retrieval: current techniques, promising directions and open issues. Journal of Visual Communication and Image Representation, 10(4):39--62, 1999.

[00142]   [30] Y. Rui, T.S. Huang, M. Ortega, and S. Mehrotra. Relevance feedback: A power tool in interactive content-based image retrieval. IEEE Transactions on Circuits and Systems for Video Technology, pages 644--655, 1998.

[00143]   [31] Y.S. Kim, W.N. Street, and F. Menczer. Feature Selection in Unsupervised Learning via Evolutionary Search. In 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-00), pages 365—369, San Diego, 2000.

Figure 1

Figure 2

negative example　　　　positive example
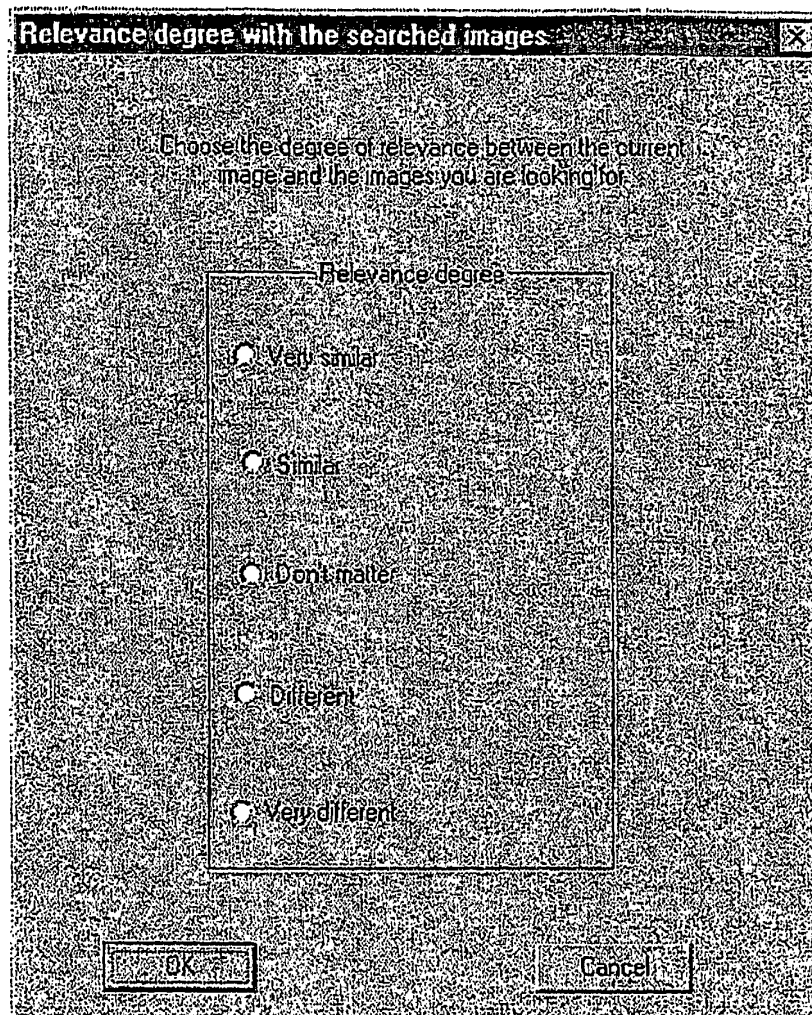

Figure 3

negative example　　　positive example
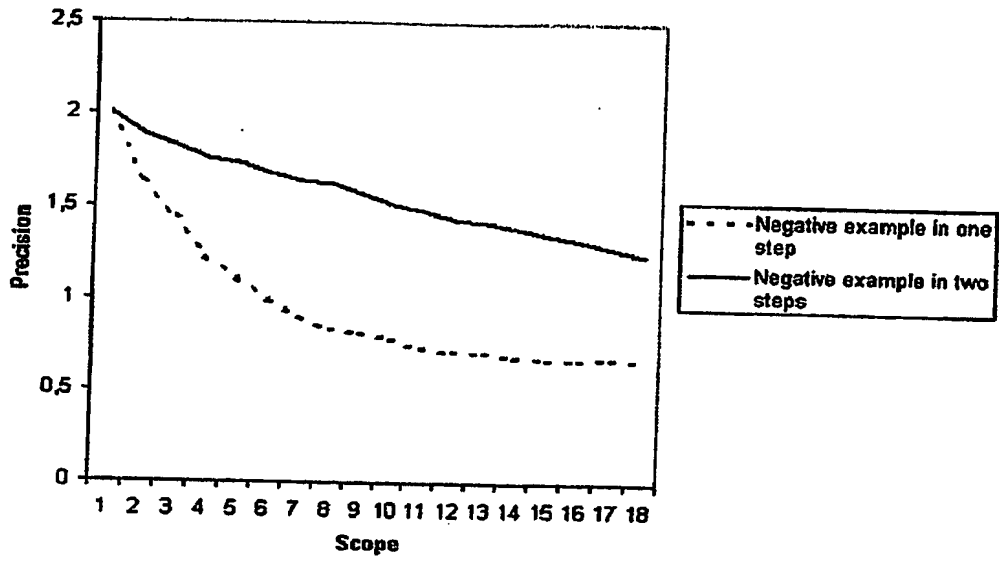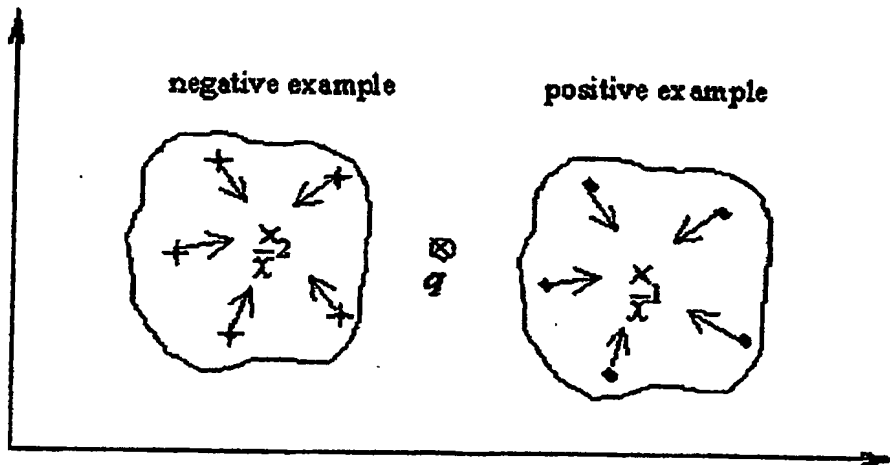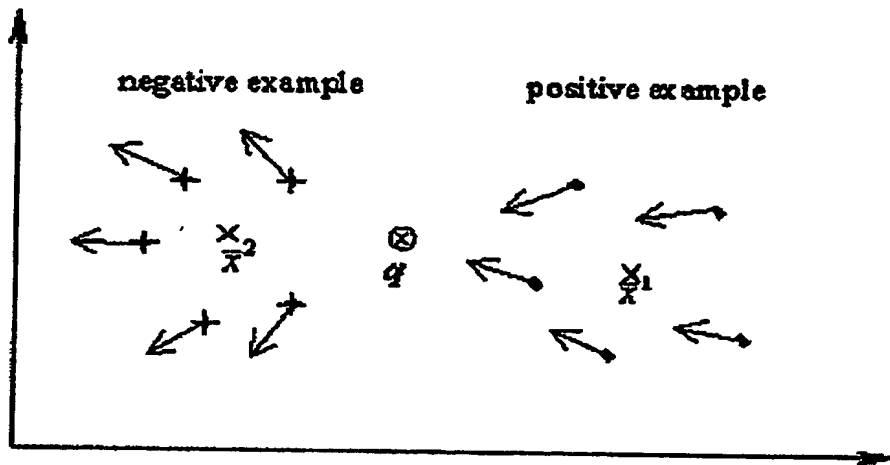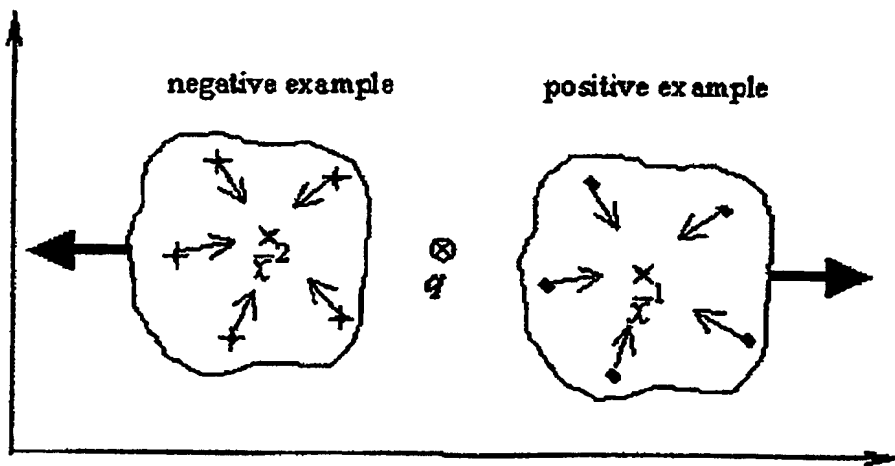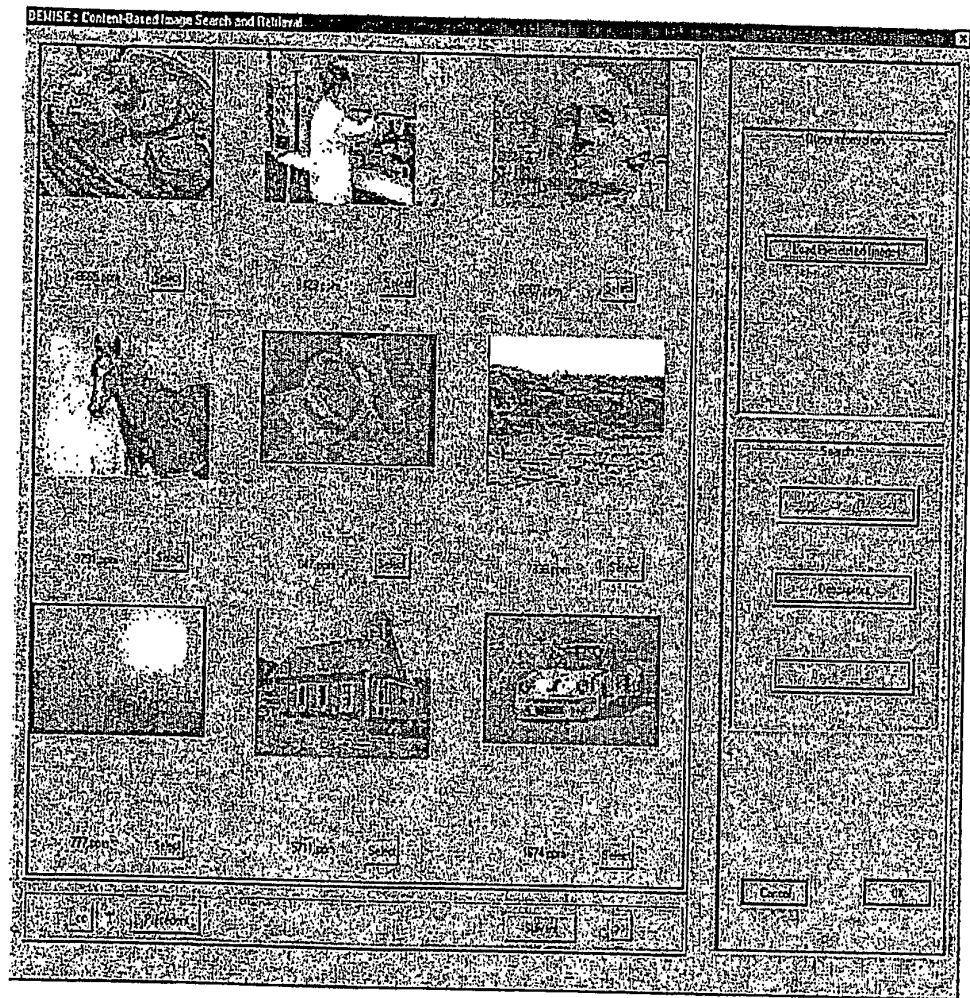
Figure 4

Figure 5

Figure 6
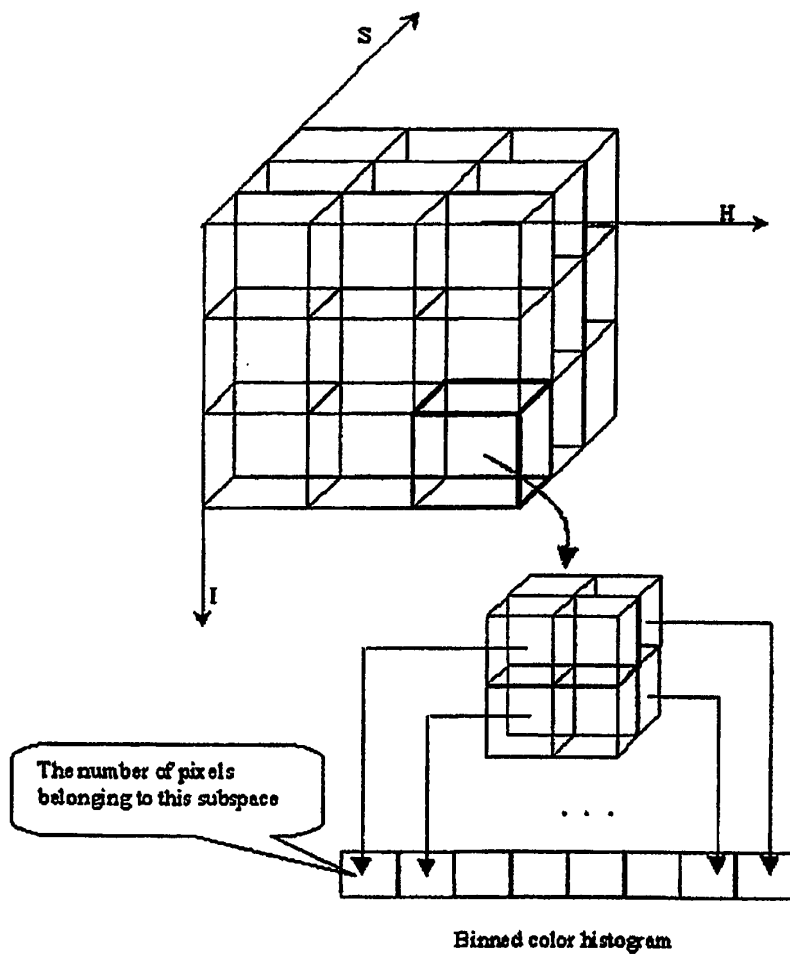
Figure 7
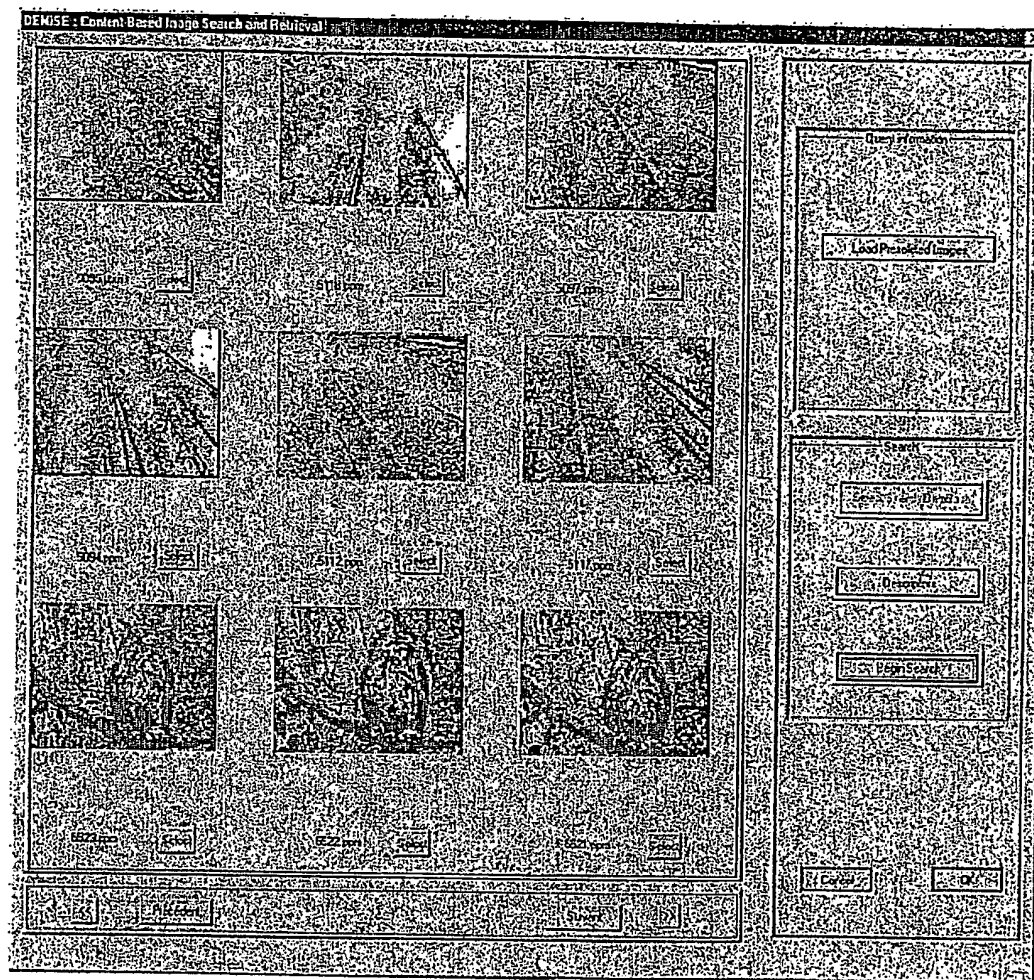
Figure 8
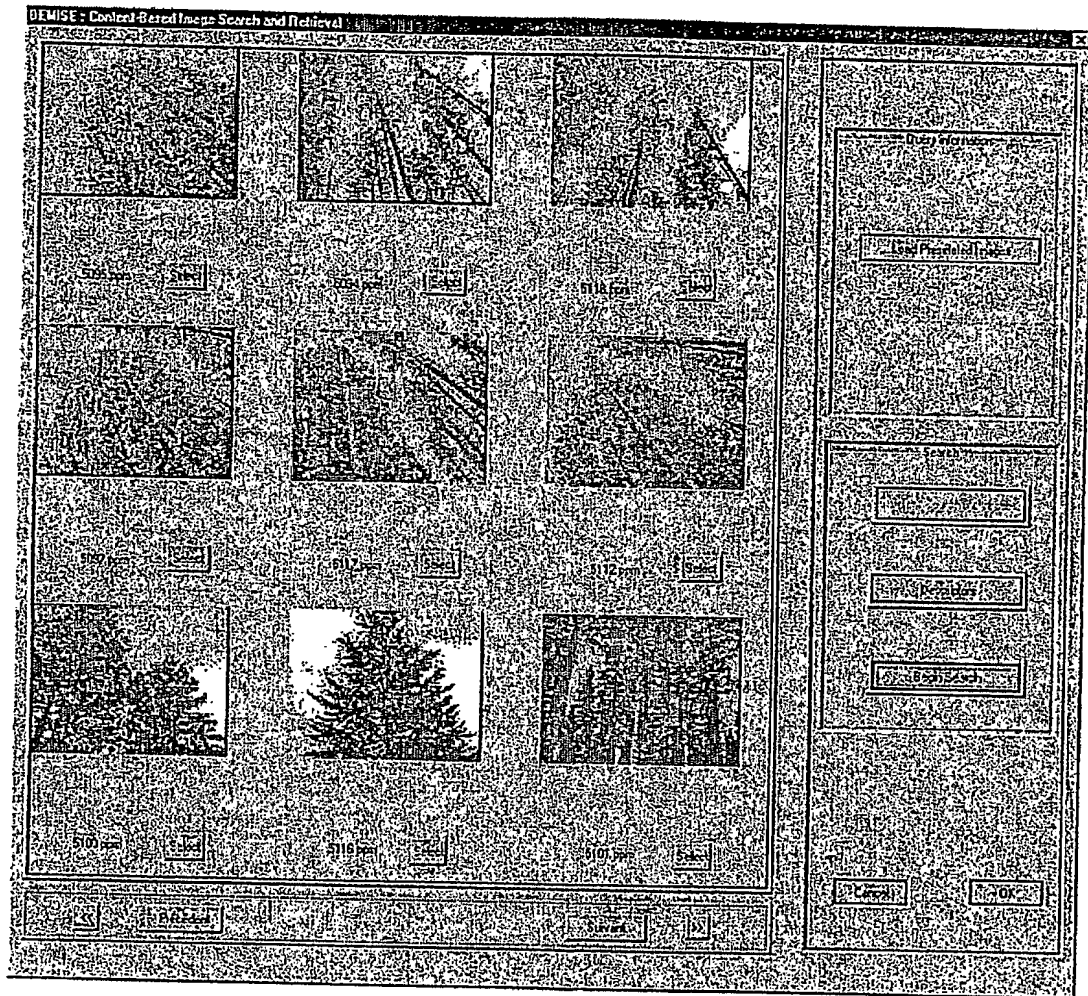
Figure 9
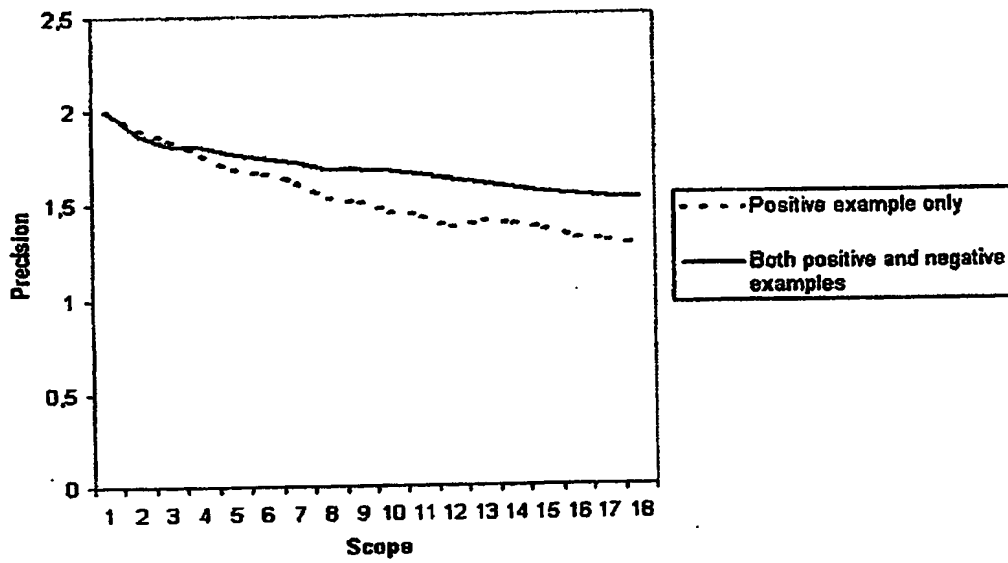
Figure 10

Experience 2.1

	S1	S2	S3	Average
Only PE	2	4	3	3
PE and NE	1	3	1	1,67

Experience 2.2

	S1	S2	S3	Average
Only PE	fail	fail	12	fail
PE and NE	3	12	1	5,33

Experience 2.3

	S1	S2	S3	Average
Only PE	12	9	5	8,67
PE and NE	2	4	3	3,00

Experience 2.4

	S1	S2	S3	Average
Only PE	12	9	fail	fail
PE and NE	3	4	3	3,33

Figure 11